Release 1.0

88-8 17 August 1988

WHY ARE EXPERT SYSTEMS FRIENDLY?

Any self-respecting commentary on expert systems always takes care to point out how friendly they are, as if that were technologically inherent in their nature. In fact, expert systems are friendly because (and only when) people bother to build them that way; we've seen in-house demos that answer most "why?" questions like this: "Because you said so."

We're continually nonplussed at the mystical reverence expert systems evoke among people who don't build or use them. Yes, they are more than just programming even as a computer is more than metal and silicon, or a house more than wood and glass. But they're still built by people and controlled by people. There's no more in there than what some person put in.

What's in there is a data model. Early-style expert systems described the data in rules; more efficient, larger-scale expert systems use databases or (object-oriented) frames rather than assert static data as rules. And then there are rules qua rules, designed to deal with dynamic data. If A is true, then B is true. If B is true, then C is true. And so forth. You can have combinations of situations that will make other situations true -- or false; and you can say something is a certain percent likely to be true. Then there's the inference engine, which applies the rules against the defined data and the new, dynamic data that sets off the inferencing.

That's all you really need for an expert system. For a commercial expert system, of course, you need much more. You need a user-friendly interface, with questions to elicit the data and text files to provide explanations. Why did it ask this question? How did it reach this conclusion? Showing the user the actual sequence and text of the rules fired may frighten him away. ("If user_name equals not_expert and system_rules greater_than_30, then do user_is_confused.") Finally, you will also need links to other sources of data and to other applications.

So why are expert systems friendly? Not because the technology makes them so (to the contrary). They are friendly because by the time they were implemented commercially, developers had learned to make their systems

====>

COME SEE US AT AAAI BOOTH 1029

INSIDE

FRIENDLY SYSTEMS 1
Why?
NITTY GRITTY EXPERTS 3
De Kleer's brute-force TMS.
jaNETor, an earnest proposal.
Start-over at Symbolics.
ExpertiCe in Saber-C.
Grammar, release 1.1.
IMPLEMENTATION SUPPORT 13
Beyond the decision.
NOT BY UNIX ALONE 16
CICS on a stick.
RELEASE 1.1 -- FEEDBACK 17
Groupware revisited.
HYPERACCESS 19
Many ways to skin a Mac.
FREE ADVICE 20
Desktop publishing.
RESOURCES/CALENDAR 21/22

EDventure Holdings Inc., 375 Park Avenue, New York, NY 10152, (212) 758-3434

usable by real people. (Much of the computer industry is now coming around to this way of thinking and stealing some of the "AI" vendors' thunder with higher-level representations of data and logic in object-oriented databases, groupware and user interfaces.) Expert systems have traditionally been built explicitly to interact with people, who supply the dynamic data that sets off the inferencing. This projected user community is not dp personnel or even lowly clerks who could be forced to learn a confusing system, but rather higher-level people whose time is costly. Finally, most expert systems are built to express explicit business rules and policies, rather than to implement lower-level procedures dreamed up and designed by a programmer. They deal with objects and concepts -- people, crankshafts, product flaws, company units -- rather than data-processing primitives.

Back to the technical world

On the other hand, the tools for designing expert systems aren't necessarily so friendly. (Just use one!) Moreover, expert systems are starting to operate in a less friendly realm. That is, expert systems will increasingly share data with traditional dp sources, at a far higher bandwidth than communications with people. Indeed, the greater part of the effort to develop American Express's Authorizer's Assistant (**Release 1.0**, 86-10, 87-9) was spent on connecting it to AmEx's databases, rather than on the expert system itself or even the user interface. Ascent/System One's GateKeeper (**Release 1.0**, 88-6) includes not just an expert system, but extensive portable database facilities built around Oracle's dbms. The development of expert systems will become just one more programming exercise, a task assigned to people familiar with CICS, database and other dp commonplaces. This doesn't mean, of course, that expert systems should become any less friendly, but that they will act additionally as a bridge between the two worlds (jaNETor, page 5) or will operate embedded within a system where they rarely communicate with end-users. Since one of the benefits of expert systems is their ability to represent decision-making logic and real-world concepts clearly, the best embedded expert systems will remain accessible to human beings who may wish to inspect or change the logic. Only if no one were ever to look at it or change it again, would an expert system work more effectively hard-coded in a traditional cryptic low-level form.

Meanwhile, the kind of friendliness based on user interfaces (rather than underlying functionality and intelligibility) will provide limited commercial advantage in the future, not because of diminishing need, but because of increasing supply: Friendliness will be taken for granted. For their part, expert systems vendors will do better not to reinvent the wheel, but to adopt and enhance interface standards. If/then Solutions uses 1-2-3 as a vehicle for a demystified low-end expert system tool; vendors such as Neuron Data and IntelliCorp are adopting Windows for their PC-based systems; Neuron Data is reselling display tools from V.I. Corporation.

Thus the role of the expert system tool vendor will evolve. No longer leaders in user interface, some ES developers will become pure tool vendors, enabling people to build functionality into applications where they are components, not the foundation. Others will develop market- or function-specific expertise and products, as Coherent Thought is doing for diagnosis (**Release 1.0**, 88-1), Gensym has done for real-time systems, IntelliCorp has done for data modeling and simulation (87-1), and Symbolics is doing for graphics and animation (page 8).

NITTY-GRITTY EXPERTS: BRUTE-FORCE TRUTH MAINTENANCE

One of the more interesting papers at this year's AAAI conference will be Mike Dixon and Johan de Kleer's "Massively parallel assumption-based truth maintenance." The paper outlines their experience putting de Kleer's assumption-based truth-maintenance systems (ATMS) onto Xerox PARC's recently purchased Connection Machine. As it happens, the Connection Machine didn't bring much new to the concepts of assumption-based truth maintenance, but it made it easier and faster to implement them. Of course, that may change... Meanwhile, the Connection Machine's data parallelism shows great promise of simplifying implementation issues so that researchers can concentrate on basic issues rather than tactics -- and, yes, so that real-world problems can be solved more quickly.

At PARC since 1979, de Kleer has spent much of that his working on truth-maintenance systems (TMS), large frameworks for managing the maintenance of "truth" -- a consistent set of beliefs that retains its integrity in the face of changing assumptions and justifications. (An assumption is a belief or fact; a justification is a link between two or more facts, i.e. a rule or inference.) In effect, the approach divides many problem solvers into two components -- one concerned with the specific rules of the domain, and a more general TMS concerned solely with managing the current state of the search. The problems with early TMSes were many -- an inability to maintain multiple, contradictory sets of hypotheses and a need for constant recalculation to reset the stage each time new data comes in and validates or invalidates specific subsets of the "truth." The TMSes were cumbersome but not inherently complex; the challenge of building them was due to the vast number of possibilities that made efficient strategies difficult to devise given the limits of traditional serial processing.

One such strategy was *assumption-based* truth maintenance, a new approach for better organizing the search through the TMS space of alternatives or hypotheses that minimizes redundant effort and logical dead-ends. This problem-solving technique was developed by de Kleer and his colleagues and was rapidly incorporated into such products as KEEworlds in IntelliCorp's KEE.

The ATMS explicitly represents the structure of the search space, with locations in the space representing assumption sets. But it only instantiates the relevant parts of it -- assumptions and links to new beliefs based on the inferences the system has made. It is this superior ability to limit activity to only the relevant sets of assumptions by storing the dependency links between assumptions and beliefs that distinguishes the ATMS. It treats many assumption sets not as unique sets but as combinations of sets which may already have been instantiated (branches of a tree, metaphorically). Common information is shared across contexts, or sets of potentially conflicting hypotheses; when new data is added, updates can automatically propagate across the system and create new assumption sets but only subsets of the search space (branches) needs to be directly manipulated.

Naturally, even this approach takes heavy calculation to deal with the combinatorial explosion of assumptions and justifications. Just four true or untrue assumptions, for example, result in 2^4 (16) possible combinations. (Not all the possible combinations are ever discretely represented.) While the logic of the ATMS still limits the total number of assumption sets maintained at any one time, the Connection Machine ATMS no longer needs to explicitly represent the links between assumptions and beliefs, because it simply keeps all the relevant, discrete sets of assumptions in memory. There's no need to follow a set of links, because the parallel ATMS can be updated directly by a single process which goes on simultaneously across thousands of processors and thousands of possible sets of assumptions.

(Some assumption sets are lumped together until/unless a relevant justification makes the distinction between A1 and A2. There's one set of assumptions per (virtual) processor; the processor dies when they become inconsistent or irrelevant. When a single set of assumptions bifurcates with the addition of a new assumption (into A1 where it is true and A2 where it is not), the process divides and creates a second processor. In fact, most of the possibilities -- sets of assumptions -- are inconsistent and need not be represented.)

THE CONNECTION MACHINE

Most commercial parallel machines allow you to run many processes in parallel, and therefore require you to decompose a problem into separate processes that can operate simultaneously. Frequently different processes may act on the same data in sequence, requiring careful attention to issues of synchronization. Programming these systems can be quite complex unless there are several discrete modules to a program. In that case, the parallelism may not offer much more than good communications among several separate machines and effective load-sharing -- and of course high performance.

By contrast, the Connection Machine (see Release 1.0, 88-1) is quite simple. Each of its 16K to 64K processors has its own data. (PARC's has 16K processors and 128M of memory in total.) All the processors perform the same function in lockstep, each acting on its own data. The CM can create virtual processors at runtime, so that it can be configured dynamically to suit the changing (data) dimensions of any problem it is handling, thereby simulating any data-parallel machine.

For problems that involve a large and changing amount of data, this approach is ideal. As it turns out, truth maintenance is such a problem. Rather than calculate the impact of a new fact by propagating the change through a series of forward- or backward-chaining calculations, as in a traditional TMS, the new Connection Machine ATMS manages a huge in-memory database of hypotheses (combinations of assumptions), and addresses each directly with new information. Each processor that holds an assumption affected by the new information updates itself accordingly.

On the other hand, the ATMS makes little use of the CM's internal communications, raising the possibility that the CM is simulating an application-specific parallel system that could perform the same function -- and that function only -- at a fraction of the cost of a general-purpose Connection Machine. Such a back-end system (so much for the swing to standard hardware!) would be well-suited for data- and constraint-intensive problems such as truth maintenance, as well as scientific visualization, machine vision, parsing, diagnosis and constraint-based modeling.

Suddenly the complexity of truth-maintenance vanishes, speeding up both the design and execution of the solution immeasurably. It took only four days to put together an algorithm and make it work, Dixon and de Kleer note with delight. With it, one constraint problem -- with minimum problem-solving logic but a lot of ATMS work -- took 45 seconds on the Connection Machine, vs. 70 minutes on a Symbolics...with an inverse ratio for cost, to be sure!

Changed assumptions with the CM approach

The implications of using the Connection Machine are more interesting for speeding up AI research than for the content of the research per se. The ATMS problem has already been addressed, ably. "In fact, the CM version is closer to the original abstract ideas," says de Kleer. "You don't benefit from the complex [serial] approach; it's just tougher to build." Researchers will be able to concentrate more on basic strategies and algorithms rather than clever implementation and coding techniques designed for a resource-constrained world.

In the future, use of processors such as the Connection Machine will dramatically speed up the rate at which new ideas can be implemented, tested and refined -- or abandoned. And it may also change the way people think about their problems. "To date," says de Kleer, "we've implemented the same problem-solving strategies on the CM. But now we have all the possible solutions arrayed in front of us. We might want to explore other control strategies [that have nothing to do with the prior inference sequence]. We can see which processors have the minimum number of assumptions, or the highest probabilities or contain certain assumptions of particular interest, and focus on those. Now we can make the search space [not just its structure] explicit, and query all the possibilities directly."

* * *

jaNETor: AN EARNEST PROPOSAL

If expert systems are so smart and so friendly, why can't they make other software look smart and friendly? Maybe they can. We describe such a (hypothetical) system below. Why are we giving this idea away for free? One, because it's the implementation, not the idea, that has value. And two, if somebody would make it, we'd like to use it -- please!

The biggest problem facing most customers and most vendors of computer systems is the one not even mentioned in the ads (because it would be negative and the other guys don't) -- management of the system itself. The simplest solution is the one Apple used in the original Mac: limit the options so you limit the decisions the user must make. The second is the one everyone claims to use: implement standards so that everything works seamlessly together (but there are too many standards!). The third: send in or hire a systems/network administrator who has taken classes from each of the vendors in question and who may be able to integrate and manage the systems with a little bit of luck and a huge amount of perseverance.

If someone gave us $5 million to create an AI startup for the broadest possible market, we would eschew a generic expert-system tool (even one with a graphical user interface and database connectivity) or even a system tailored for Wall Street, where the money flows freely if you can generate re-turns (doesn't it?). Instead, we would address this most common, most intractable problem faced by the vendors and users of all these systems, installing and managing the computer environment itself.

The solution is not a high-level systems administrator with arcane technical knowledge, but rather a simple, humble product called jaNETor^{tm}. It keeps the system functioning at peak efficiency, maintains the premises, and moves things around on request. To limit the scope somewhat, we will address the market for pc-based network management, a large subset, with an OS/2- or UNIX-based version of jaNETor. As it happens, there is little that jaNETor doesn't have to take care of, since operating systems, peripheral devices and applications all rely on smooth operation of the network. (Didn't someone say, "The network *is* the computer"?) jaNETor leaves off at edge of the network, where it delivers messages to the appropriate gateway or protocol converter for communication to mainframe and mini-based systems. A future version will be more global, with facilities to handle dp center operations. And there will be a higher-ranking expert called coordiNETor, in charge of setting up work groups and managing workflows. (See page 17.)

Easier said than done...

What's keeping VCs and entrepreneurs from jumping in to create jaNETor?[1] Although this is fantasy, the vendor that can come up with a tool to automate tasks that most people are still doing painfully one at a time will be the Microsoft of the Nineties (maybe it *is* Microsoft).

Theoretically, developing jaNETor should be an achievable challenge, since you're dealing with specific, explicit software systems with specific interfaces. In practice, aside from the issue of poorly designed or incorrectly specified interfaces, the problem is one of combinatorial explosion. Each network has its own APIs; each peripheral has its own drivers, a minor problem, and its own way of specifying them, a major problem.[2] We'd settle for a standard way of *specifying* interfaces, given the lack of standard interfaces, but even that is unlikely in this ever-moving, ever-improving market.

jaNETor has to keep up with everything important on the market -- OS/2, Presentation Manager, all the LANs in all their versions, all the important applications (just selecting them is tough enough!); there is no way to be both complete and up to date. Of course, that's what network administrators try to do, but with only a subset of the array that jaNETor must handle.

Then there's the almost infinite variety of user configurations and the need for network tuning -- still an art, not a science, dependent on applications and usage patterns. And that's still not all. Industry licensing practices generally force users to buy and install their OS separately from their network or application software, for example. (3Com, for one, has solved this problem by contracting with Microsoft to bundle DOS with its network software -- a benefit of its involvement in the Microsoft LAN Manager.)

---

[1] *Many of these facilities are showing up piecemeal and in a form that still requires extensive system administrator attention. User profiles (with pre-loaded applications and data), disk management and other facilities are commonplace -- but not automatic.*

[2] *With the Micro Channel, IBM has at least taken the first step by turning installation into a software rather than a hardware problem, as long as the third-party vendors follow IBM's rules.*

...but worth doing

With those caveats, let's consider the functions of jaNETor. jaNETor plays the part of the network and systems administrator,[3] letting users stick to the specific content of their tasks. We don't mind learning the command for global replace or automatic initial upper case if we use those functions, but why should we have to set up a new subdirectory? If there's a defined way to do it, why can't the software do it? Well, jaNETor can.

The first thing jaNETor does is to install itself, according to what the system user tells it. "What operating systems will you be using? What kinds of hardware?" jaNETor's knowledge of the specifics of a wide variety of environments is its primary value. Each of these problems can be solved easily enough in isolation, with enough resources brought to bear, but only jaNETor can handle any combination of tools and applications as it comes up. Thus jaNETor combines the capabilities of a protocol converter (from many to many) and system configurer. It can manage devices by system, logical or generic names ("any line printer on the third floor"). It can manage security and access, and provide requisite accounting information to vendors who license per user or per session. It can communicate with the 20 most popular "standard" end-user productivity packages so that cumbersome installation and communication routines will become a thing of the past.

Of course jaNETor automatically allocates disk and other resources. It monitors the use of these resources and informs users of likely problems. "Your server is now at 80 percent of capacity. Would you like to select some files for archiving? Here are three subdirectories that have not been touched in the last six months which are taking up 13 percent of capacity. They were all created by Fred Bloggs, who has not logged on since 19 October 1987." Now there's a smart jaNETor!

Adding applications is easy. jaNETor asks for the name and version number of the application, and which users should have access to it. Then jaNETor creates subdirectories for each user that will automatically be selected each time that application is loaded (from a menu of choices). When a user decides to use a Paradox file in a Xywrite mail-merge task, he can query jaNETor for a list of Paradox files, not for "e:\par\sub\cust.lis" or whatever the lower-level name might be. If necessary, jaNETor can even clean the files using a utility purchased from a third party.

jaNETor, needless to say, runs under OS/2 or UNIX on a server. It understands DOS and the Mac OS and a variety of flavors of UNIX. It can also negotiate with TCP/IP, OSI and standard IBM mainframe environments, although in its first release it does not attempt to do everything. It is not a universal operating system; it is a facility that configures systems, monitors them, and performs cross-system tasks on demand. Because it knows so much, it is very large and is delivered on CD-ROM.

---

[3] *Although jaNETor's vendor will work with other vendors, it will not ask them to support jaNETor; jaNETor will work with or around their products just as any network administrator would (although jaNETor's developers, who are network experts par excellence, may make a few humble suggestions).*

START-OVER AT SYMBOLICS

When we told a friend we were on our way to see Symbolics, he asked, "Is there anything left to see?" [Or to buy?] That's the kind of crack that makes new ceo Jay Wurts wince, and it's one of the company's biggest problems. In fact, if Symbolics were a start-up, people would marvel at its technology. As it is, the company has limited resources and faces the delicate balance between cutting back enough to stay alive, and spending enough to have something worth living for in the future.

Jay Wurts, co-founder and ultimately president of Management Decision Systems, sold the company for $47 million to Information Resources in 1985 and was thinking of starting up a new company to leverage the coming boom in fax with a strategy for the transition to e-mail in the future. But then he noticed Symbolics. As a man who made the switch from engineer to manager, he was intrigued with the possibilities Symbolics represents and was uniquely well-equipped to understand both its problems and its potential.

Under Wurts, Symbolics now understands that the future lies largely with standard hardware, but its customer base and its product line still center on proprietary hardware. The trick is to accomplish the transition gracefully, without sacrificing (more of) the people and the software expertise that are its greatest assets -- $11 million in the bank notwithstanding.

For now, the company's biggest problem is the skepticism of customers and potential customers, of which our friend is typical. On the pr side, Wurts has met with 350 customers (100 companies) since he arrived last May. On the action side, he has performed some product triage and cut staff sharply, as did several managements before him. Total staffing is down to 450 from the 1986 peak of 975, and he has focused attention on completing the surviving products, some announced last summer at AAAI and shipped late or yet to be delivered.

Now there are fewer products in the pipeline, and more coming out at the end. The development tools announced in April (Release 1.0, 88-4) are shipping now. The company has been shipping the 386 UNIX version of CLOE, a build-on-Symbolics/deliver-on-stock-hardware system, since last April, and is announcing the MS-DOS version this week to ship in September. The Ivory LISP chip products, announced at last year's AAAI for shipment late in 1987, are going into beta. The first of them, MacIvory, a co-processor for the Mac, will not address as large a market as the follow-ons, a co-processor for the PC and a VME-bus standalone version, but it will prove the concept, and will start generating some revenues by year-end.

Buy now! Don't wait!

Next is the issue of making the Symbolics software available on other platforms without reliance on co-processors -- not something the company wants to preannounce, given that it's still trying to sell the current versions. "It's important for people to realize that they can already get real value from it right now," says Wurts. "We don't want to encourage them to wait" -- or especially to buy other people's hardware while they wait.

Symbolics provides powerful tools for expert systems (Joshua), text management and documentation (Concordia) and object-oriented database (Statice) around its rich development environment (Genera). Customers are reluctant to use these tools because they're afraid of being locked in, so that resolving the standards issues has implications beyond the amount of revenues generated from standard-hardware versions themselves.

All of the company's current efforts work toward surmounting the standards hurdle. CLOE allows you to develop on a Symbolics 3600, with all its facilities, and deliver on standard hardware using Common LISP. The new tools work (as does Genera) on an Ivory chip tied to standard environments, as demonstrated in an impressive setup illustrating the integration provided by Concordia, Symbolics' hypertext document management system. You can use it to manage Mac files (documentation and supporting illustrations, say) across a network of Macs, some with and some without Ivory co-processors. Concordia points to the relevant Mac files, and always brings up the latest version for display or print-out. Texts or data in Concordia can also be integrated into an expert system, so that a user can find information through inferencing rather than through a direct hand-built link.

Let's get specific

Long-run, Wurts wants to bring in more vertical-market expertise, along the lines of the company's animation/graphics division (notwithstanding plans to sell Macsyma, its mathematics tool). A key factor in that move is Joshua, an object-oriented, extensible expert system tool which makes it simple for users -- or Symbolics itself -- to design expert system shells targeted at particular problems by supplementing Joshua's object classes and problem-solving procedures with ones specialized for specific kinds of problems.

Although the market is definitely moving towards standard hardware, it is also moving towards the kinds of capabilities in which Symbolics has unique expertise. Such areas include scientific visualization, CASE, design functions ranging from 3D and constraint-based modeling to fluid-flow analysis and simulation, and high-end expert systems. All these markets are being addressed at the low end on standard platforms, but at least some customers will want to use high-end solutions if they see a painless path to low-end versions in the future. At the same time, chips such as the Ivory and a general trend to multi-vendor environments are lessening the hardware issue, which now is more a question of perception than of practical difficulties.

The world is going Symbolics' way. Some of its products may prove harder for other suppliers to emulate than is now apparent, and customers may turn to Symbolics for "stop-gap" solutions for years. By the time they become available elsewhere, some customers may have discovered that the Symbolics solutions already work fine.

However, history may prevent the company from exploiting that trend. The question is, Will Symbolics still be ahead of the pack at that time? There is controversy within and outside the company over the sharp cuts in spending on future products. Many difficult design problems that Symbolics tackled early on are now being addressed elsewhere, in some cases by talented people who have left Symbolics (among them Dan Weinreb, a key figure behind Statice; Jan Walker, a key figure behind Concordia; hardware architect Jack Holloway, vp of technology; and Macsyma co-developer John Kulp, vp of R&D). Wurts clearly feels it's more important to buy credibility and revenues now, and use the funds and presence gained thereby to catch up and enter new markets by acquisition in the future.

EXPERTICE

Those who love LISP don't love it for itself alone, but for the glorious full-service development environments that come with it. Yet such capabilities are not inherently linked to LISP; that's just an historical accident. Now this kind of environment is starting to turn up around other languages, which may offset one of the primary factors in LISP's favor.

Saber Software, the commercialization of a Harvard computer science thesis, is typical of the companies that populate the virtual garages of Cambridge. In the summer of 1985 Sesha Pratap and three classmates wrote a C interpreter; the following winter Pratap went to work for venture capitalist Wolfensohn & Co.; the following summer he left to turn the tool into a product and build a company, Saber (one of the founders is a fencer).

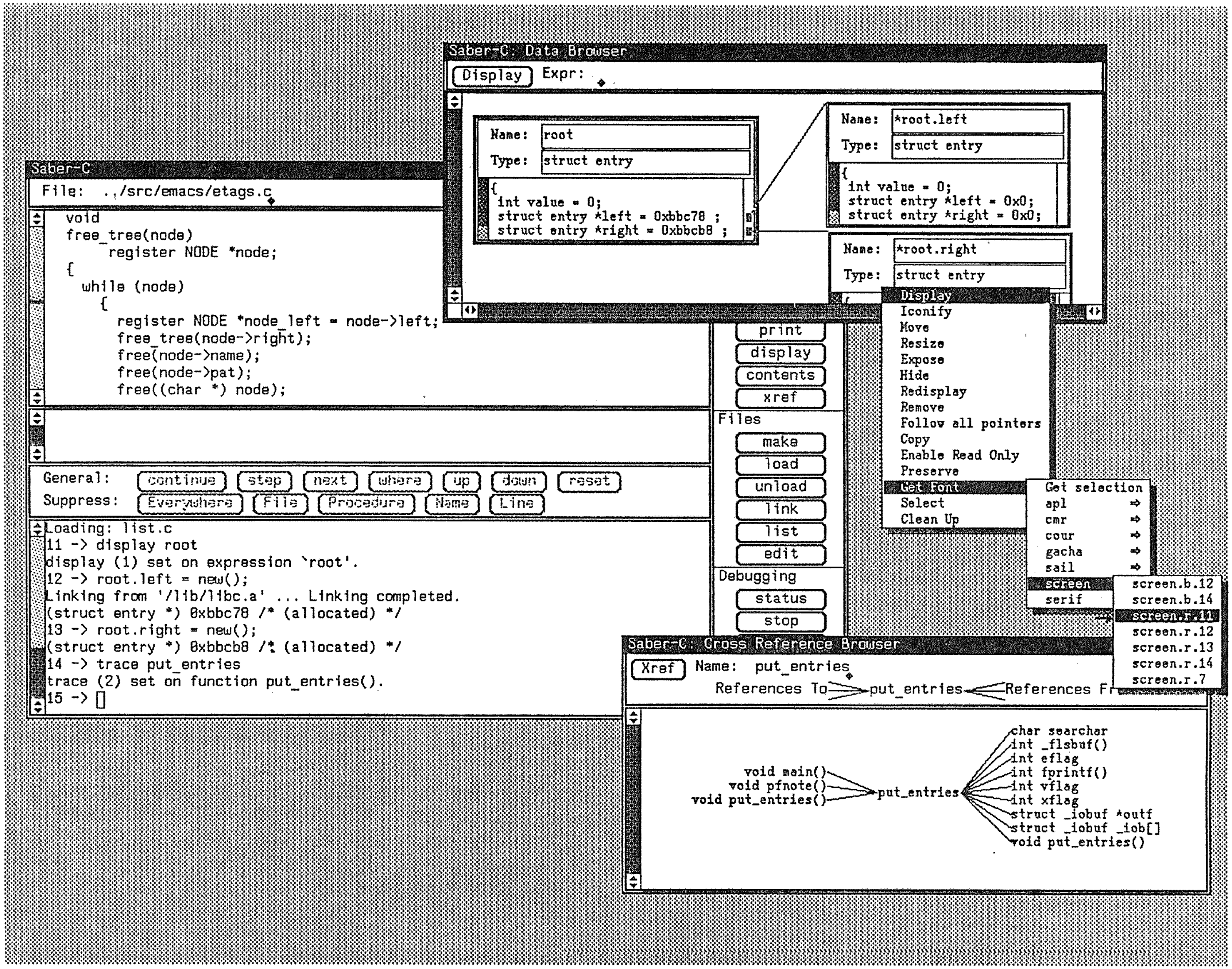
Saber-C was announced at UseNIX in June and has already garnered 250 users, at $1995 for a single copy. The product makes a Sun look like a LISP machine, with windows, browsers, tracers and the like (see across). Although it performs many of the functions of a debugger, and can be used at the end of the development cycle for QA, its value lies in its ability to detect errors before it's too late -- when the code is still in fragments and protoype modules, rather than in relatively complete, compilable form. (The system contains 130,000 lines of C, developed with its own help, and 100 lines of assembler.)

While debuggers generally first compile the source code, thereby losing a lot of information, Saber-C interprets C code and detects runtime errors as the code executes. It flags not just errors of syntax but of semantics, such as references to out-of-bounds pointers, inconsistent memory allocation and incorrect array indices. When errors occur, Saber-C pops up an edit window with an error message that explains precisely what the problem is: Not just "pointer out of bounds" but "pointer went bad on line 42; pointer = 0x01000, low bound = 0x00100, high bound = 0x00900; pointer used to point to variable: test." If you select the word "test," Saber-C will pop up the point in the program where that variable was defined.

Saber-C can display functions that call or are called by a line of code or a phrase, or a linked list of data structures; if you select one of these elements from the graph, it will call up the appropriate lines of code. You can also create conditional statements. For example, you might want to see what happens to a program only when certain values are exceeded, or when certain conditions are true.

Integration

Saber-C combines and integrates the capabilities of a number of tools you could probably buy separately, and in so doing makes all of them more effective. The user ends up with the ability to look at his code from a number of angles, to filter it according to any number of criteria, to pick through it by following data or execution flows. Long-run, we could see the integration of Saber-C with a CASE front-end environment. Given its ability to understand the code it is managing, it could probably integrate at a very detailed level: That is, rather than pass files from one system to the other, the combined tool could manage the relationship between discrete, small-grained design units and the code that implements them.



Name: root
Type: struct entry

```
{
 int value = 0;
 struct entry *left = 0xbbc78 ;
 struct entry *right = 0xbbcb8 ;
```

Name: *root.left
Type: struct entry

```
{
 int value = 0;
 struct entry *left = 0x0;
 struct entry *right = 0x0;
```

Name: *root.right
Type: struct entry

File: ../src/emacs/etags.c

```
void
free_tree(node)
     register NODE *node;
{
  while (node)
    {
      register NODE *node_left = node->left;
      free_tree(node->right);
      free(node->name);
      free(node->pat);
      free((char *) node);
```

```
Loading: list.c
11 -> display root
display (1) set on expression `root'.
12 -> root.left = new();
Linking from '/lib/libc.a' ... Linking completed.
(struct entry *) 0xbbc78 /* (allocated) */
13 -> root.right = new();
(struct entry *) 0xbbcb8 /* (allocated) */
14 -> trace put_entries
trace (2) set on function put_entries().
15 -> []
```

Name: put_entries

That same ability to derive substantial meaning from C source files means that Saber could also be the first step towards a reverse-engineering tool, a field that is exciting venture capitalists and code-entwined customers alike (see **Release 1.0**, 87-10). The market for reverse-engineering in C is not as large as that for COBOL (with some 80 billion lines of congealed spaghetti). On the other hand, the task is more challenging, since C code tends to be sparse and hard to understand (which is why C is often called a high-level assembler in disguise).

The sum of all these features is a compiler-independent design environment rather than an after-the-fact debugger. Saber-C is ideal for prototyping and for catching errors when they're still cheap to fix because of its speed and its ability to handle code fragments as well as entire programs. CASE tools notwithstanding, most programmers still prefer to start with code segments rather than diagrams; Saber-C (like LISP) lets them do so with a minimum of irrevocability.

To conclude, we can't resist quoting one of Saber-C's users, Scribe author Brian Reid. He posted this comment to an electronic bulletin board: "...About once every five years some new technology comes along that changes my life. My first bicycle. My first car. Federal Express. My first home terminal. PostScript printers. And now, Saber... If you don't use Saber to write your next C program, you're a dork."

C Refinery

Meanwhile, Reasoning Systems is about to ship its C Language Subsystem (a name no marketing man could ever love) for $1900 on a Sun or Symbolics machine. CLS will allow C programs to be moved into the Refine knowledge base, a LISP-based program storage and inspection environment originally developed to manage software in Reasoning's own Refine language. (A Refine-to-C compiler will be available next year.) The knowledge base stores detailed program structure and data, which could be used as the foundation for a variety of content- and syntax-sensitive tools such as Saber-C. In other words, CLS is more general, but does less.

Symbolics also has a C inspector, but the company as yet lacks the requisite market expertise to sell it successfully. As we said above, CASE is one of the vertical markets that Symbolics may explore.

RELEASE 1.1 (SPEAKING OF LANGUAGE-SENSITIVE ENVIRONMENTS)

If you think parsing C is a challenge, try English! Last year we saw an early version of Houghton-Mifflin's Grammar Checker, and liked it so much that we invited the company to present it at the PC Forum. The product is now ready and marketing plans disclosed: DEC will resell it as the VAX Grammar Checker. What makes the job so hard -- and VAX Grammar Checker's achievement so great -- is that it not only parses "correct" sentences, but can make a good guess as to what is wrong with incorrect ones (a myriad of possibilities once you remove the constraint of good grammar). But we have to wonder what the checker would do with this sentence, in a letter from Digital itself: "I wish to thank all of you for the participatory consideration you gave to our recent teleconference." For further details and some puns, see **Release 1.0**, 87-10.

IMPLEMENTATION SUPPORT

As we've pointed out previously, direct manipulation and interactivity are just a way of letting the user do all the work -- unless the process itself is the goal. Thus in desktop publishing (for example), vendors are looking for ways to further automate the process with pagination rules, expert layout tools and the like. Likewise, there's the notion that decision support will turn into expert systems which will make wise decisions automatically. But that's mostly wrong-headed. A more promising concept is implementation support -- a broader vision that treats decision-making as just one more step in the process of executing strategy -- a process that includes defining assumptions, making decisions, and carrying out the decisions with the help of informed, enthusiastic people.

The key to decision implementation is full communication -- not just of the decision itself, but of the assumptions underlying it and the roles people will play in executing it. Software can be a valuable tool in that communication -- helping people to build the perspective and assumptions that will ensure their understanding and commitment once the decisions are made. While you don't need software to make a decision, you can use it to help explain one.

But in practice people usually use it the wrong way: Decision-making is not desktop publishing, where the product is the output. Decision-making is a process -- which includes getting people to understand and sign up for a decision as it is made...and perhaps even to contribute to its content in the most enlightened operations. In this case, software is not a tool to create documents in a paper medium; software *is* the medium.

We call this market *implementation* support: When an executive makes a decision, how does he get his people to sign up for it and carry it out? By letting them in on the process. In the cheap sense, implementation support is just a fancy name for a set of software tools, as decision support is for spreadsheets. But it also implies a way of using those tools that is as broadly meaningful as factory automation -- far more than the purchase of a couple of machines.

From recipient to participant

Implementation support means getting people to share a vision of the world, not just of the discrete factors that directly affect a decision. This doesn't happen by letting people vote or by showing people charts, but by getting people to help *build* the charts, in a process Palo Alto, CA, consultant Bernie DeKoven calls computer-enhanced meetings. Indeed, implementation of a decision properly begins before it is made, making the process inherently a group effort. Thus implementation support tools are used to manage a group arriving at a consensus -- even one already determined by the group leader. (If he can't get the group to rally round, maybe he should re-examine the decision!)

Aside from DeKoven's work (box), the clearest example of the idea we've seen is the visual-interaction program at Coopers & Lybrand's Strategic Management Services group. Consultants, after all, are in an awkward position: They generally can't make or implement decisions; they have to persuade their clients to do so. (The best internal managers behave as if they were similarly constrained, even though they are not.)

At Coopers & Lybrand, implementation support is a carefully planned process, involving both a meeting leader and a computer-wielding facilitator who have worked together preparing materials for the meeting. These vary widely from meeting to meeting, depending on the subjects discussed, but they generally include models of the business under discussion -- internal resources and costs, external conditions, information flows, org charts and matrices, etc. The difference, says SMS director and program leader David Braunschvig, is

TECHNOGRAPHER EXTRAORDINAIRE

One of the leading figures in implementation support is Bernie DeKoven, a Palo Alto consultant who started out as a teacher and gravitated to the game business. He discovered computers when Atari sent him a VCS and he realized that its power lay not just in the game itself, but in the user's ability to determine the rules of play. "Setting up the rules and renegotiating is as important as playing," he says. His best-known book, *(Bernie)'s Rules of Order*, is actually a template (hence the punctuation) for people to use in defining their own rules of collaboration. By the time they have worked together to define the rules, they are probably comfortable enough to work together on the problem.

DeKoven went on to design computer games for Mindscape, Epyx and Children's Television Workshop. He gravitated from games and schools to business customers when he discovered, "The schools' management structure was antithetical to what I was trying to teach: You have to collaborate to design games. Unbeknownst to them, businesses are in fact collaborating and are very responsive to tools that help that -- because it helps get results. So I can nurture that in businesses and I may eventually get back to education and government."

He has set himself up in business to train people in "technography," the art of facilitating meetings with a pc, usually a Macintosh, to elucidate people's statements and ideas. But the training is not just for specialists: Someday the participants will wield mice themselves, and an electronic mediator will keep the users from overwriting each other's contributions. Towards that end, DeKoven also helps customers build their own "group computing labs," or combination drop-in center, computer training and meeting facilities -- the next buzzword beyond information centers? The difference is that ICs work on the "library" model of one person/one machine; the GCL encourages social interaction and the use of computers as communication tools.

As befits a practitioner, DeKoven is eclectic in his choice of tools, and can work with almost anything the customer (typically a company trying to spur a team effort) has on hand. By choice, he frequently uses MORE and GrandView from Symantec, and Decision Pad, a weighting/ranking tool from Apian Software. "Just figuring out the weights and factors with Decision Pad is a very clarifying process, whether or not you ever make a decision," he says. "Virtually anything with interactive graphics, starting with MacDraw, is a marvellous tool. You can clarify ideas, show relationships, compare things, position things."

that this information is not presented in a bound paper report (that comes later); it is prepared in half-finished form on an explicitly transient, modifiable medium -- a screen.

Thus the meeting participants join in the task of finishing the information. (They are by definition not a passive audience; Coopers & Lybrand limits these sessions to 8 to 12 people.) For example, the consultant may present a matrix (or circle chart or whatever) that represents the client's current view of the marketplace. Then he runs through the questions that his consulting group wrestled with: Does it make more sense to segment the market by processor type or by operating system? Do the same people buy both Dom Perignon and Miller Lite? What would happen if Juan's customer service department began providing feedback to Alice's product development group? What are the alternatives to distributing the conference across three separate locations: different dates, different place, fewer people?

As the group considers these questions, the facilitator manipulates the computer -- a Macintosh, in C&L's case -- to reflect the discussion. Suddenly the shapes on the chart change; the boxes on the org chart shuffle themselves. The arrows showing product or information flow shift direction; the numbers in the spreadsheet recalculate to reflect new assumptions. If two clear sides emerge, the facilitator may divide the screen into two, with points and counterpoints. Presenting the arguments in plain text tends to lessen inhibitions but also diminishes the unfair advantage of oratorical firepower, notes Braunschvig wryly.

A sense of community

Of course, it's not always easy. The meeting leader and the facilitator must be expert in their subject; their ideas must make sense. The technology is a tool, not a replacement for traditional consulting skills. But when it works, the system works spectacularly well, says Braunschvig: "We are different from the Japanese; we're heterogeneous. People come with mixed perspectives that might otherwise build barriers. We have to compensate and this technology helps us share a common platform." The meeting participants understand the discussion and contribute to the outcome. They are not presented to; they join in creating a set of models of the problem they're about to handle.

The technology plays a key but small part in all this. In traditional meetings, minutes may be distributed a couple of days after the fact, long after the meeting is over and its messages forgotten. The minutes tend to provide the text of decisions formally stated, rather than explanations of the models and assumptions underlying them. At C&L, the documents are printed out hour by hour, for participants to scribble on and take with them when they leave. And the information they contain is clearly presented -- courtesy of the facilitator's skills and of Macintosh-based presentation tools such as Symantec's MORE, Microsoft's PowerPoint and Aldus's Persuasion -- all more typically used to make static slides and overheads.

The value comes when the audience gets into the act -- when the tools are used not for static presentations such as an artist could draw, but to bring a group together in interactive creative thinking.

UNISOFT'S UNITECS: CICS ON A STICK

Formerly nicknamed the Berkeley Port Authority, UNIX porting house UniSoft has now moved next door to Emeryville and is attempting to redefine itself as a full-service systems software house. It has ported countless versions of UNIX onto a variety of Motorola 68000 hardware platforms, including Mac, and is just completing a port onto the AMD 29000 RISC system. At the same time, it is broadening its array to offer a full suite of systems software to its OEM customers, including device drivers, a full range of compilers and a set of connectivity tools. The most intriguing of these is UniTecs, an emulation of the popular IBM transaction-processing environment.

UniSoft was founded in 1981 by Jeff Schriebman, a former UCBerkeley instructor who is now the chief technical person at its Nippon UniSoft unit. Funded by Saunderson Holdings and Venture Link of the U.K., UniSoft helped hardware vendors adapt UNIX to their needs, while database vendors such as Oracle, RTI and Unify built up an infrastructure of more SQL sites than exists in the IBM world (IBM's creation of the language notwithstanding). UniSoft continues to broaden its array of tools. Anything you can do in the IBM world, you can do in the UNIX world, and probably cheaper. Now that even includes CICS, IBM's Customer Information Control System. CICS is a heavy-duty system which takes up several megabytes on an IBM mainframe, and just wasn't a priority as UniSoft focused on development tools.

In 1987, UniSoft's U.K. arm contracted to convert an IBM-based manufacturing system to UNIX. It turned out that in the process the group implemented CICS in the UNIX environment without recognizing the ostensible magnitude of the task. UniSoft has worked to turn this capability (unique so far as we know) into a product, shipping now to OEMs such as Pyramid and installed at five user sites.

CICS is somewhat the opposite of UNIX: It's not considered a standard by governments or standards bodies, since it is sold for profit by IBM without any blather about openness, but it's widely used by mainframe sites. Its presence on UNIX will make it much easier for customers to move their existing mainframe transaction applications to a UNIX environment with a minimum of effort. They can also keep some of the application(s) on mainframes, and distribute other functions to UNIX boxes -- in branch offices, say. A reseller could offer both IBM and UNIX versions of the same transaction application, depending on customer size and preference. The system can also easily convert VSAM files (EBCDIC) to UNIX format (ASCII).

For those who know and care, UniTecs emulates the latest version of CICS, 1.7, and offers about 80 percent of the functionality of the IBM original, with error recovery, table-based security, and operates in both conversational and pseudo-conversational mode. The major gap, automatic timing (so that the system itself can start up a transaction at a specified time), will be filled in release 2. The system does not support 3270 terminals, but it can make its own ASCII terminals look and feel like 3270s.

The success of third-party vendors such as UniSoft and UNIX utilities house Unitech is both an indicator of the growing success of UNIX in the commercial market and a contributor to it. While the value-added may reside in applications in the long run, in the short run it resides in the tools that will make those applications portable and possible.

RELEASE 1.1: GROUPWARE REVISITED

We recently received a letter from Tolly Holt, founder of Coordination Technology, commenting on and amplifying the discussion of groupware and CT's future product "Scaffold" in our June issue (88-6). Holt stresses the notion of virtual workspace wherein people perform work with a specified set of tools, work products and people, a concept to be reified in Scaffold. Consider this a work paper on issues to be addressed, rather than questions already answered. Here are some edited excerpts, followed by further comments from us:

Holt:

...some remarks from your friendly coordination mechanic about your structured-information/structured-transactions axes:

In every work environment, electronic or not, there are two fundamental types of motion: (a) people moving from place to place; and (b) operational effects propagating from place to place. The latter often takes the form of work products moving from place to place.

"Places" in a work environment are defined by function. Therefore, the pattern of people motion (a) is driven by their preferences or requirements as to which function to exercise when; the pattern of operational effect propagation [or paper-shuffling] (b) is driven by how the various work functions relate to one another.

Products whose main point is so-called information-sharing will tend to provide a structured information space in which people can move about, driven by their changing interests. Access controls can place limits on this movement.

On the other hand, products whose main point is to support the interdependence among people and functions may confine each person to a relatively small information space and concentrate on the transmission of work products (messages, documents, code modules, forms) from person to person. The products assume that each person can resolve the issue of which of his interests or work functions is involved.

No work environment tool can succeed if it does not support both types of motion to some degree [or provide the extensibility to let the tool-user build in such support].

...How, typically, do the effects of work operations propagate from place to place in the information-sharing model? Essentially via persons as carriers. Take as a typical example group document creation using hypertext. Individuals clambering about through the information space modify it as they go. The texts in the space have various implicit or explicit functions relative to the whole -- carrying a point to be made within the document, comments about some aspect of the document, etc.

What about the movement of persons in the context of products focused on supporting functional interdependence relations -- say Staffware or The Coordinator? The information space accessible to each individual has enough structure so that he can shift his focus of attention from conversation to conversation, from form to form, i.e. shifting from interest/role to interest/role. Where may a given person operate? Normally, in conversations in which the person has a part, or on forms with which the person is concerned. Thus products of this type, by their nature, are more explicit about what is and what is not relevant to particular persons' functions.

These thoughts are opening steps towards an enriched taxonomy, springing from the fact that the work environment has (a) multiple people; (b) multiple functions; and (c) multiple types of objects involved in function-to-function connections. It is then natural to ask about a product (or model): How are people related to functions (one-to-many, one-to-one, many-to-many mappings)? How are functions related to functions (by cardinality mappings, via message or form transmission, tightly coupled interactions with or without person participation, by monitoring ability, etc.)? How are functions related to object types? It is further natural to ask: How does a product support work environment changes, such as changes in people/roles, functions, object types, and their relations to one another?

Release 1.1:

In essence, Holt is arguing for a more explicit definition of "work spaces" and the objects within them than is contained in the two products cited -- Staffware and The Coordinator -- and that is already promised in his own product, which we have named Scaffold since it has no name of its own. Scaffold contains facilities for defining these objects/parameters explicitly, whereas they exist only implicitly in Staffware and The Coordinator and must be constructed by the builder/user if at all. The Coordinator allows for conversations and can assign people to groups, but it does not explicitly link functions, roles and people. Staffware has its own database and sees the world as consisting mostly of transactions plus the data they are about. The follow-on product, Calypso, may well have a more explicit view of the multiple virtual "work spaces" suggested by Holt, as does our own coordiNETor (page 5).

As we said in June, the whole process of building groupware tools requires us to examine and understand and render explicit a myriad of unrecognized patterns and structures that govern our daily work. (The rules that govern extraordinary events -- with the exception of nuclear power-plant disasters and the like -- are less in need of explication and automation, although they may in fact be clearer because they are noticed for their very rarity.) By making them explicit and supporting them in groupware tools, we make it easier to build systems that faithfully mirror and automate actual work flows, and that effectively perform their functions.

RELEASE 0.1 -- JUAN & ALICE ON TURNAROUNDS

Juan:	Why did Frankenstein create a monster?
Alice:	Because he was working with used parts.

HYPERACCESS -- MANY WAYS TO SKIN A MAC

We have complained that the Mac lacks an "industrial-strength" database management system (Release 1.0, 88-6). However, there's no real reason that these databases need to exist on the Mac as long as users can get to them easily from a Mac (although it means less revenue to Apple). Now there's a proliferation of competing approaches, including two from Oracle alone.

Oracle has announced its long-promised implementation of Oracle for the Mac (for shipment by year-end). But it has done more than simply port its dbms to the Mac: It has done so behind a HyperCard interface, based on "Hyper*SQL." Although Hyper*SQL was only a minor part of the total effort, it's a notable development. It can be embedded into HyperTalk so that one can use HyperCard as a front-end to an Oracle database on a Mac or (with proper communication facilities) other machine. Hyper*SQL is not magic, of course, and someone has to write the HyperTalk scripts to use it, but it can be a very handy tool. Through these facilities, a HyperCard user has access to the latest information from a server database, extending the one-by-one data entry and direct-manipulation features of HyperCard to provide access to a more powerful back-end. However, the Mac version of Oracle supports only a local user and can't operate as a server to any other system, a function of the limitations of the Mac OS.

Oracle is also working to embed Network Innovations'/Apple's CL/1 connectivity language into Oracle, so that any Oracle database can support the dozen or so Mac front-end applications that will shortly use CL/1. (Oracle for the Mac will be the first Mac-based back-end, as opposed to front-end, to support CL/1, although only for applications on the same physical machine.) CL/1 is a more general language than Hyper*SQL that can be embedded in virtually any Mac application, and, with the CL/1 interpreter added to the target machine/database such as Oracle, manage the execution of queries on virtually any remote relational database as well as DEC RMS data files. Access Technology has already incorporated CL/1 into its front-end database browser MacDBC (for Data Base Connection), providing access to whatever databases the user may have through CL/1, as well as to other databases that Access supports (Cognos, DEC RDB and other mini databases and 4GLs). From the Access perspective, CL/1 helps Access to broaden the selection of back-ends its Mac front-end supports; from the Oracle perspective, CL/1 lets Oracle broaden the range of Mac front-ends its database supports.

Either way, Apple sits happily in the middle. This fall, Apple will release further linking tools as part of a connectivity engine from its newly purchased Orion Network Systems unit which will provide the low-level underpinnings (such as APPC) for managing the nuts and bolts (or pipes and valves) of cooperative processing among Macs and mainframes.

Finally, there's the approach of KnowledgeSet's HyperKRS, which lets HyperCard act as a front-end to HyperCard stacks which it automatically indexes. The user (or developer) need not build the links by hand, which saves a lot of work. Secondly, the system is much more easily updatable to absorb newly entered text than would be a hardwired system with static links. At the user's or builder's option, the user can select pre-built queries, such as "all containing 'La Honda'," or he can generate his own at runtime using simple Boolean expressions. Long run, HyperKRS will provide access to any text base, inside or outside the HyperCard stack space.

FREE ADVICE: DESKTOP PUBLISHING

As anyone following the story of IBM's advertising account is well aware, five companies are currently bidding on the business. Part of the trial assignment is to develop strategies for IBM in desktop publishing and in small business.

We find it ironic that IBM selected these two markets, not particularly its strongholds -- but perhaps that's the point. We think IBM probably has a good chance in the small-business market, if it can change its image. While a large company knows that it's not necessarily easier to hook up a PC than a Mac to a mainframe environment, and is already dealing with the challenge of multiple vendors, smaller businesses can't afford to do so. Thus the small company may not want to use a Mac if it expects to grow, since there's no clear upgrade path to servers, groupware and other high-end systems.

With OS/2, IBM can legitimately promise these customers a mostly compatible upgrade path -- with servers and industrial-strength databases -- as Apple cannot yet do on its own machines. As the world gets more complex, IBM can also legitimately assert that it knows how to deal with heterogeneous environments -- its own. That is the whole point of Systems Application Architecture -- the integration of disparate systems rather than the illusory appeal of homogeneity that will ultimately break down.

On the desktop publishing side, however, we think IBM is addressing a telephone-operator market (and one in which it won't be a leader to the extent that desktop publishing is defined as a personal-productivity tool). Most of the functions in current dtp tools will become a part of regular word-processors, much as teleconferencing and other functions are now part of everyone's telephone.

Functions that are not common are more properly considered *electronic* publishing, which will remain the province of specialized groups and of server applications, not the mass market Apple and its disciples and perhaps IBM are implicitly predicting. Electronic publishing comes in two parts -- presentation, which the individual does at the low end and a specialist does at the high end -- and process management, inherently a high-end application sold to a business, not to its workers, and implemented on networks and servers. While Apple can integrate publishing tasks and materials at a single user's desk using an array of connectivity tools (page 19), it takes server-based applications to manage a large-scale publishing effort. This includes such functions as managing the sequence of document and image creation or retrieval, editing, assembly, review, selection, printing, revision, importation of data and the like, in never-ending cycles.

* * *

RELEASE 1.1

Compete! from ManageWare, Inc., described in our April (88-4) issue, has just won a good housekeeping seal of approval from Arthur Andersen & Co., which will be using it as tool in its worldwide consulting business. And, yes, it can do implementation support!

RESOURCES & PHONE NUMBERS

Jeff Hulton, **Access Technology**, (617) 655-9191
Bill Ray, **Apian Software**, (415) 851-8496, (800) 237-4565
Don Casey, Mike Homer, **Apple Computer**, (408) 996-1010
David Braunschvig, **Coopers & Lybrand**, (212) 536-2640
Tolly Holt, **Coordination Technology**, (203) 268-4045
Bernie DeKoven, (415) 857-1757
Kathy Sowards, **Digital Equipment**, (603) 884-3575
Robert Moore, **Gensym**, (617) 547-9707
Robert Benson, **If/then Solutions**, (415) 322-3430
Chris Bowman, Roger Badertscher, **KnowledgeSet**, (415) 968-9888
Bill Bane, **ManageWare**, (203) 359-9397
Jim Groff, Paul Weinberg, **Network Innovations (Apple)**, (408) 257-6800
Marc Benioff, Mike Wilson, Pete Tierney, **Oracle Corporation**, (415) 598-8000
Paul Rampel, Jim Mullen, **Orion Network Systems (Apple)**, (415) 649-4000
Gordon Kotik, **Reasoning Systems**, (415) 494-6201
Sesha Pratap, Marv Goldschmitt, **Saber Software**, (617) 876-7636
Tom Byers, Gordon Eubanks, **Symantec**, (408) 253-9600
Jay Wurts, Howard Cannon, **Symbolics**, (617) 621-7500
Danny Hillis, Sheryl Handler, **Thinking Machines**, (617) 876-1111
Al Sisto, Nancy Price, **UniSoft**, (415) 420-6400 x422
Paula Brooks, **Unitech Software**, (703) 264-3301
Mike Dixon, Johan de Kleer, **Xerox PARC**, (415) 494-4398

COMING SOON...

- *Active objects. (It is expected that passive objects will also be discussed.)*
- *Database styles: Oracle, Sybase, Interbase, Cullinet, IBM et al.*
- *Graphics standards; graphics and structure.*
- *Constraint-based modeling.*
- *And much more...*

Release 1.0 is published 12 times a year by EDventure Holdings, 375 Park Ave., New York, NY 10152; (212) 758-3434. It covers the pc, software, CASE, groupware, text management and connectivity markets, and artificial intelligence. Editor & publisher: Esther Dyson; associate publisher: Lisa Braden; circulation & fulfillment manager: Lori Mariani; executive secretary: Maria Soler; copy chief & transcripts editor: William M. Kutik. *Copyright 1988, EDventure Holdings Inc. All rights reserved. No material in this publication may be reproduced without written permission; however, we gladly arrange for reprints or bulk purchases.* *Subscriptions cost $395 per year, $475 overseas.*

RELEASE 1.0 CALENDAR

August 28-September 1	**Information center conference and exposition** - Boston. With Michael Hammer, among others. Sponsored by Weingarten Publications, publishers of Information Center magazine. Contact: Nancy Murray, (617) 542-0146.
August 29-September 1	**Very large databases** - Los Angeles. With Robert Kahn. A topic looming large in the future. Contact: Arie Shoshani, (415) 486-5171.
August 30-31	**Increasing management productivity with intelligent systems** - Snowmass, CO. Sponsored by the University of Colorado's Center for Applied Artificial Intelligence. Contact: Melissa Eiles, (303) 492-8229.
September 6-10	**International neural network society** - Boston. First annual meeting. With tutorials and speeches by Carver Mead, Bernard Widrow, Stephen Grossberg, Terrence Sejnowski. Contact: J. R. Schuman Associates, (617) 237-7931.
September 7-10	**Optical information systems** - New York City. Sponsored by Meckler Corp. Contact: Hannah Preston or Marilyn Reed, (800) 635-5537 or (203) 226-6967.
September 7-10	**SPA fourth annual conference** - Washington, DC. With Gordon Eubanks, Symantec; Victor Alhadeff, Egghead (newly public; congreggtulations!); John Roach, Tandy. Contact: Jayne White, (202) 452-1600.
September 13-15	**DataStorage88** - San Jose. The annual management forum on storage technology and issues, run by storage gurus Jim Porter and Ray Freeman; also Finis Conner and Alan Shugart. Sponsored by DISK/TREND and Freeman Associates. Call Darlene Plamondon, Cartlidge & Associates, (408) 554-6644.
September 14-16	**Ashton-Tate developer conference** - Los Angeles. "A meeting of the minds." SQL Server, dBASE IV, and more. Sponsored by Ashton-Tate. Contact: Bonnie Wacker, (213) 538-7145.
September 14-17	**Seybold Desktop Publishing Conference** - Santa Clara. Sponsored by Seybold Seminars. The usual stellar line-up. Contact: Kevin Howard, (213) 457-5850.
September 15-16	**The semiconductor outlook** - Boston. Sponsored by Technologic Partners and Merrill Lynch. With Dick Shaffer, Jerry Sanders, Murray Goldman, T.J. Rodgers, Phil Kaufman, Rodney Smith, others. Call Robert Johnson, (212) 696-9330.
September 18-22	**Design Management Conference** - Martha's Vineyard, MA. Sponsored by Design Management Institute. With a cross-fertilizing group of speakers with perspective from academia to corporate and social issues: Olivetti, Royal College of Art (London), others. Contact: Melissa Blanchette, (617) 236-4165. (Sold out, but call to get on the list for next year's.)

September 20-21	**ShowCASE conference III** - St. Louis, MO. Yet another CASE conference. This one sponsored by the Center for the Study of Data Processing, Washington University. With James Martin, Ed Yourdon, Vaughan Merlyn, Capers Jones, Chris Gane, et al. Call Donna Skaggs, (314) 889-5380.
September 25-27	**Agenda '89** - Carlsbad, CA. Second annual, with all-new speakers this year. Run by Stewart Alsop, managed by Marketing Partners, sponsored by PCW Communications. Call Elizabeth Withrow, (415) 592-8880.
September 25-30	**OOPSLA** - San Diego. Object-oriented Programming: Systems, Languages and Applications. Sponsored by ACM. Contact: Allen Otis, Servio Logic, (503) 644-4242 or Barbara Noparstak, Digitalk, (213) 645-1082. (The conference section of OOPSLA is Wednesday through Friday (28-30), so you can catch most of CSCW first if you miss the OOPSLA tutorials.)
September 26-28	**SECOND CONFERENCE ON COMPUTER-SUPPORTED COOPERATIVE WORK** - Portland, OR. Sponsored by ACM. Contact: Suzanne Sylvia, (617) 225-1860.
September 26-28	**The software re-engineering symposium** - San Francisco. Sponsored by DCI. With Rich Currier, Panoramic; others. Contact: Marny Peabody, (508) 475-6990.
September 26-28	**CD-ROM Expo** - Chicago. Sponsored by IDG Conference Management. Workshops, tutorials, etc. Call Lisa D'Ambrosia, (800) 343-6474 or (508) 879-0700.
September 26-30	**Interop 88** - Santa Clara. The third TCP/IP interoperability conference and exhibition. Sponsored by Advanced Computing Environments. With Bob Metcalfe, Bill Joy and Dave Boggs of DEC. Contact: Dan Lynch, (415) 941-3399.
September 27-29	**Networld 88** - Dallas. Seminars and exhibits. Sponsored by Novell. Contact: Adam Torres, (201) 569-6406.
September 27	**Venture capital and IPO seminar** - San Francisco. Co-sponsored by Adapso and Accel Partners. Contact: Sheila Wakefield, Adapso, (703) 522-5055.
October 3-4	**Downsizing corporate systems** - New York City. Coping with decentralized computing, led by Ted Klein. Contact: Ted Lein at Boston Systems Group, (617) 423-1670.
October 3-5	**Adam Green's dBASE^R symposium** - Burlington, MA. Three-day conference of dBASE third parties, including clones, compilers and complements. Contact: Marny Peabody at Digital Consulting, (508) 470-3870.
October 3-6	**Electronic imaging '88** - Boston, MA. Hear the speeches; see the products. Sponsored and managed by the Institute for Graphic Communication. Contact: Richard Murray, (617) 267-9425.

October 5-6	**Home office computing forum** - Laguna Niguel, CA. Sponsored by Home Office Computing Magazine. Keynote by Alvin Toffler. Contact: Terry Mazza, (212) 505-3580.
October 7	**Boston Computer Museum Superbowl** - Boston. Come watch the industry's luminaries loom lightheaded, all in a good cause. Or sign up to be a corporate sponsor. With Dave Bunnell, Esther Dyson, Adele Goldberg, Dave Hathaway, Will Hearst (the "Examiner"), Bill Joy, Mitchell Kapor, Allen Michels, Casey Powell, Bill Poduska, Dick Shaffer. Contact: Gail Jennes, (617) 426-2800, x341.
October 7-9	**Hackers 4.0** - Saratoga, CA. Theme: "Connections." Forget E-mail; you have to be there in person! Sponsored by the Point Foundation. Contact: Glenn Tenney, (415) 574-0546.
October 9-12	**Breakaway '88** - Dallas. Sponsored by ABCD: The Microcomputer Industry Association. Theme: "Forging our future... together." Keynote by Ralph Nader. Contact: Bernie Whalen, (312) 240-1818.
October 10-12	**Frontiers of massively parallel computation** - Fairfax, VA. Sponsored by IEEE et al. Programming languages, neural nets, graph theory, everything connected with || computing. Contact: James R. Fischer, (301) 286-9412.
October 10-14	**Computer services and communications and information systems seminars** - Baltimore. The ninth annual, sponsored by Alex. Brown. All your favorite information-industry vendors, and the VCs and investors who feed (on) them. Contact: Donna Campbell or Leslie Engel, (301) 727-1700.
October 11-14	**Info Show** - New York City. Contact: Frank Fazio, Cahners Exposition Group, (203) 964-0000.
October 12-13	**Bioanalysis and computing conference and exposition** - Burlingame, CA (SFO airport). "Computing: Its role in bioanalysis." Data analysis, imaging, instrumentation control, etc. Contact: Norm de Nardi, (415) 941-8440.
October 16-19	**ACM SIGDOC** - Ann Arbor. Documentation gains cachet in a world of on-line, CD-ROM, hypertext and icons. Keynote: Ed Yourdon. Contact: Stephanie Rosenbaum, Tec-Ed, (313) 995-1010.
October 16-19	**ADAPSO MANAGEMENT CONFERENCE** - Dallas. The software and services industry's premier gathering. Contact: Sheila Wakefield, (703) 522-5055. Followed directly by...
October 19-20	**...Vardex** - Dallas. The value-added reseller & dealer exposition. Co-sponsored by Adapso and NAVAR, the National Association of VARS. Vertical markets and tailored software galore. Contact: Dan Dembicki, (313) 274-8590.

October 17-20	**C++ conference** - Denver. The first full C++ conference, sponsored by USENIX, with topics and speakers inherited from last year's C++ meetings and workshops. Chaired by Andrew C++ Koenig of AT+T Bell Labs. Contact: Judy Desharnais, (213) 592-1381 or 592-3242.
October 17-21	**Electronic document management: Approaching the 21st century** - San Jose. Sponsored by CAP International. Contact: Kristin Fisher or Jean O'Toole, (617) 982-9500.
October 18-20	**Open software '88** - London. With speakers from Sun, IBM, AT&T, etc. Sponsored by Sphinx Software. Contact: Amanda Stuart, (44 62) 882-2266.
October 23-27	**American Society for Information Science** - Atlanta. The 51st annual. With speeches by Charles Marshall, AT&T; Max Hopper, American Airlines; Anthony Craig, GEISCO. Contact: Bob Hurd, (202) 462-1000.
October 23-28	**Monterey Classic** - Monterey, CA. Where investors and companies discuss common goals... Contact: John Baumeister, (408) 987-4200.
October 24-25	**Electronic messaging '88** - Boston. Keynote: A rare appearance by Digital's Ken Olsen. Sponsored by the Electronic Mail Association. Contact: Mike Cavanagh, Louise Spieler, (202) 293-7808.
October 24-25	**Intertainment '88** - New York City. A conference on interactive entertainment for and by vendors in software, broadcasting, movies, videotex, CD-ROM, theatre, advertising, etc. Keynote: Nolan Bushnell. Also: Gary Arlen, Trip Hawkins, Robert Gehorsan (Prodigy), "Tamara," Ed Schlossberg. If you don't know these names, forget it; if you know them all, you must be a dilettante. Sponsored by Alexander & Associates. Call Sally Chin, (212) 382-3929.
October 24-27	**Conference on software maintenance** - Phoenix. Sponsored by IEEE and other organizations. Talks by Robert Arnold, Software Productivity Consortium, among others. Contact: David Card, (301) 589-1545.
October 24-28	**Federal computer conference and Defense & government computer graphics conference** - Washington, DC. Sponsored by National Council for Education on Information Strategies. Contact: Luanne Bell, (800) 343-6944 or (301) 961-6575.
October 25-27	**Teamwork, technology, & organizational performance** - Cambridge, MA. The sixth annual Seybold Executive Forum. Contact: Deborah Hay, (617) 742-5200.
October 25-28	**Educom '88** - Washington, DC. Campaign for excellence: Education, government, industry. With Eric Bloch, NSF; Alan Kay, Apple; Ross Perot, himself. Contact: Evelyn Thompson, (609) 520-3343.

October 26-28	**Venture forum '88** - Boston. Sponsored by Venture Economics, the authority in the field. Talks by many VCs and by perennial supplicant Bill Poduska (Prime, Apollo, Stellar, ...?). Contact: Diane MacArthur, (617) 237-1121.
October 27-29	**Northeast Computer Faire** - Boston. Sponsored by the Interface Group, for small systems users. Contact: Andrew Shapiro, (800) 325-3330 or (617) 449-6600.
October 30-November 3	**IIA twentieth anniversary convention** - Washington, DC. Keynote speaker: Arno Penzias, Bell Labs. Contact Mike Atkin, (202) 639-8262.
October 31-November 2	**Autofact/SME** - Chicago. Plenary speakers: John Sculley, Ken Olsen. The annual factory automation event, sponsored by the Society for Manufacturing Engineers. Contact: Jill Vanderlin or Becky Alsup, (313) 271-0023.
October 31-November 2	**UNIX Expo** - New York City. Managed by National Expositions. Contact Don Berey, (212) 391-9111.
November 3	**MICROPROCESSORS '89** - San Jose. Architects with vision from each of the major microprocessor vendors gather to discuss design, architecture and strategy. Sponsored by Microprocessor Report, an excellent newsletter. Contact editor Mike Slater, (415) 494-2677.
November 2-4	**Adam Green's dBASE^{R} symposium** - Woodland Hills, CA. See October 3. Contact: Marny Peabody, DCI, (508) 470-3870.
November 2-4	**CASE user conference** - Monterey. With Vaughan Merlyn and user war stories. Sponsored by CASE Research Co. Contact: Elizabeth Skowronnek, (206) 453-9900.
November 9-10	**UNIX: Mainstream or myth** - New York City. With Yankee's Howard Anderson and Nina Lytton; the OSF president, whoever it may be; Larry Dooling, AT&T; Jim Bell of X/Open; Max Hopper, American Airlines. Sponsored by the Yankee Group. Call Melinda Stoddard, (617) 367-1000.
November 14-16	**Design's role in customer-based business structures for the nineties** - San Francisco. With Tom Peters. Sponsored by Design Management Inst. Call Earl Powell, (617) 236-4165.
November 14-16	**The software re-engineering symposium** - New York City. Sponsored by DCI. Contact: Marny Peabody, (508) 475-6990.
November 14-18	**Comdex** - Las Vegas. The one and only, sponsored by the Interface Group. Contact: Jane Wemyss, (617) 449-6600.
November 15-17	**Neural networks and their applications** - Nimes, France. A practical-minded workshop. Sponsored by a variety of mostly French, mostly academic organizations. Contact: Norbert Giambasi, (011) (33.1) 47.80.70.00.

November 28-December 2	**International conference on fifth-generation computer systems** - Tokyo. Sponsored by Institute for New Generation Computer Technology (ICOT). Contact: Ami Semba, (011) 813 456-3195, or Hideo Asiso, (011) 8144 631-141.
December 5-7	**Strategic issues forum** - Cambridge. Sponsored by Decision Support Technology. Contact: Donna Kacin, (617) 354-6400.
December 5-8	**CASExpo** - Anaheim. Sponsored by Arthur Young; chaired by Howard Yudkin, president of the Software Productivity Consortium. Contact: Rhoda Canter, (202) 956-6041.

1989

January 17-19	**Computer graphics New York** - New York City. Targeted to end-users. Sponsored by Exhibition Marketing and Management Co. Contact: David Small, (703) 893-4545.
February 28-March 2	**UniForum** - San Francisco. Keynote by Terry Lautenbach, senior vp and gm, IBM USA. Sponsored by /usr/group. Contact: Ed Palmer, (408) 986-8840.
March 6-10	**Fifth IEEE conference on artificial intelligence applications** - Miami. Contact: IEEE, (202) 371-1013, or Mark Fox, (412) 268-3832.
March 13-18	**Seybold Seminars '89** - San Francisco. The place to be published...er, seen. Contact: Kevin Howard, (213) 457-5850.
March 14-16	**Interface and World Congress on Computing** - New York City. Moved from Chicago, in search of more enthusiasm. Sponsored by Interface. Call Walt Heithaus, (617) 449-6600.
March 19-22	**EDventure Holdings Personal Computing Forum** - Palm Springs, CA. With the first annual EDwards for EDventurous performance. Contact: Lisa Braden, (212) 758-3434.
March 28-30	**Fourth CD ROM conference** - Anaheim. Sponsored by Microsoft. Contact: Min Yee, (206) 882-8131.
April 30-May 4	**CHI '89: Conference on human factors in computing systems** - Austin. Sponsored by ACM/SIGCHI and a host of other groups. Contact: Claudia Raun, MCC, (512) 338-3798.
June 11-14	**Expert Communications '89** - San Francisco. Sponsored by Graphics Communications Association. Contact: Norman Scharpf, (703) 841-8160.
August 22-26	**IJCAI-89** - Detroit. The international version of AAAI. Sponsored by the American Association for Artificial Intelligence. Contact: Claudia Mazzetti, (415) 328-3123.

Please let us know of any other events we should include. -- Maria Soler

SUBSCRIPTION FORM

Please enter my subscription to **Release 1.0** at the rate of $395 per year in the U.S. and Canada. Overseas subscriptions are $475, airmail postage included. Payment must be enclosed. Multiple-copy rates on request. Satisfaction guaranteed or your money back.

Name________________________________

Title ________________________________

Company ________________________________

Address________________________________

City ________________ State ________ Zip ________

Telephone ________________________________

How did you hear about Release 1.0? ________________

____

AAAI8 88-8

Please fill in the information above
and send with your check payable to: EDventure Holdings Inc.
375 Park Avenue, Suite 2503
New York, NY 10152

If you have any questions, please call us at (212) 758-3434.

Lisa Braden
Associate Publisher